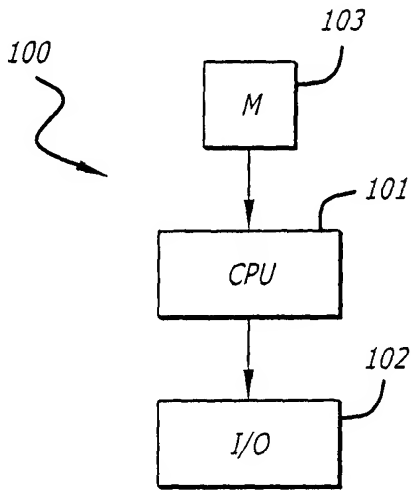
100
103
M
101
CPU
102
I/O

FIG. 1

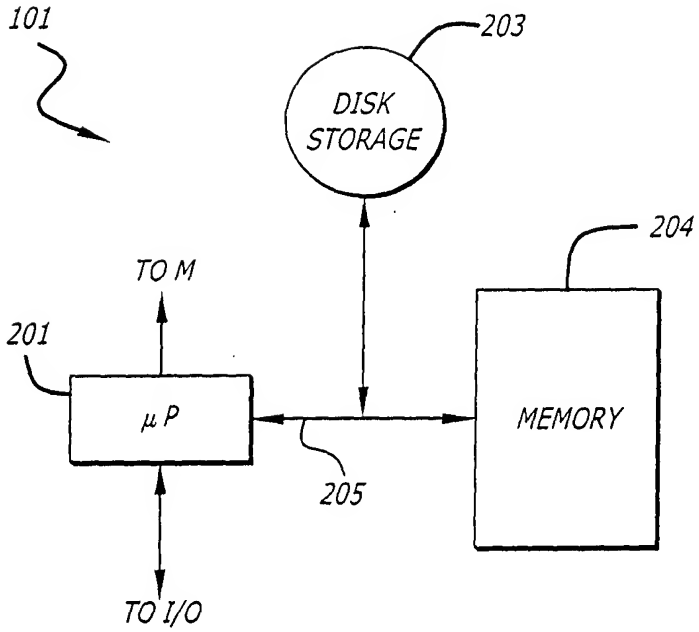
101
203
DISK STORAGE
204
TO M
201
μP
MEMORY
205
TO I/O

FIG. 2

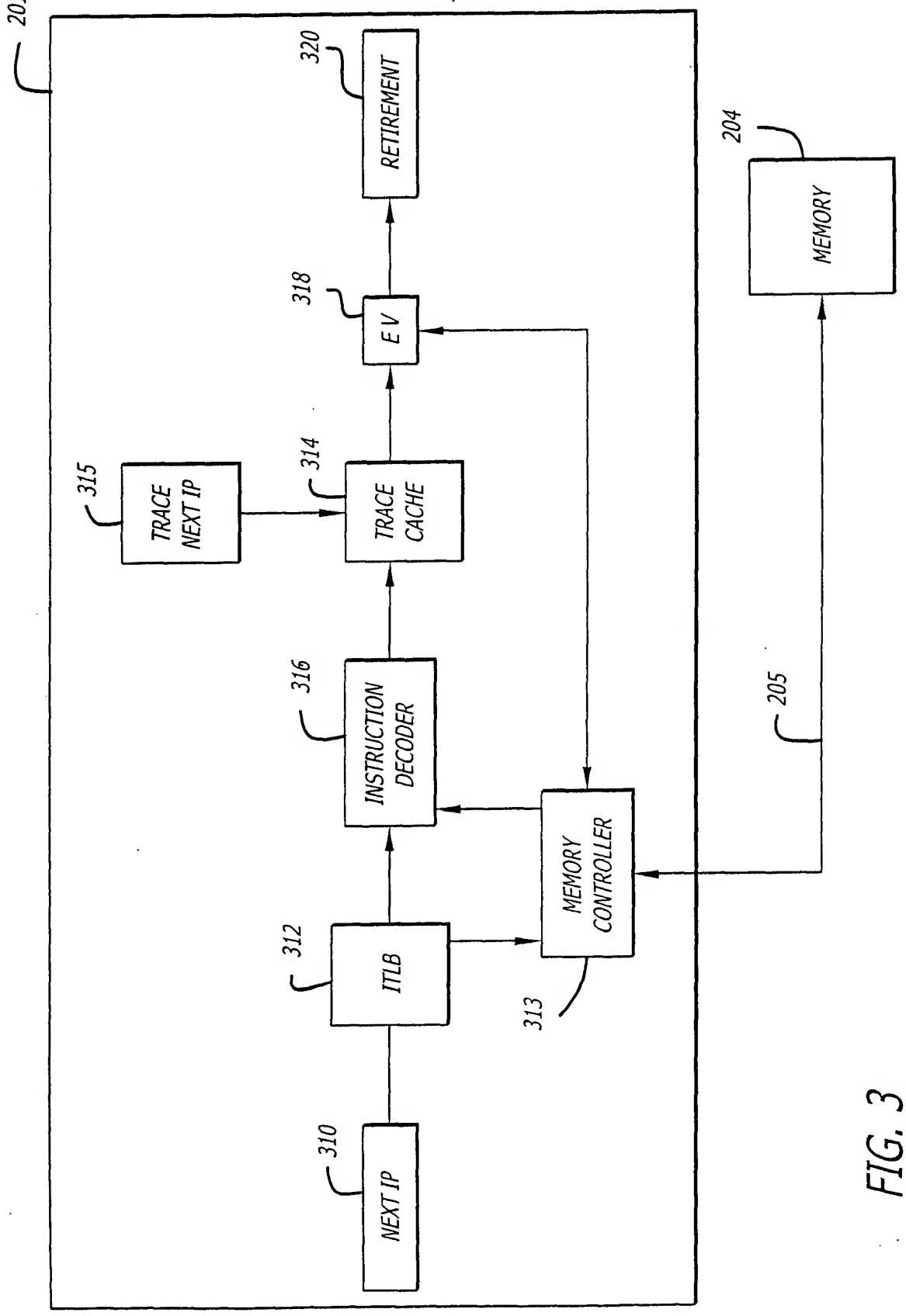
201
320
RETIREMENT
204
MEMORY
318
E V
315
TRACE NEXT IP
314
TRACE CACHE
316
INSTRUCTION DECODER
205
MEMORY CONTROLLER
312
ITLB
313
310
NEXT IP

FIG. 3

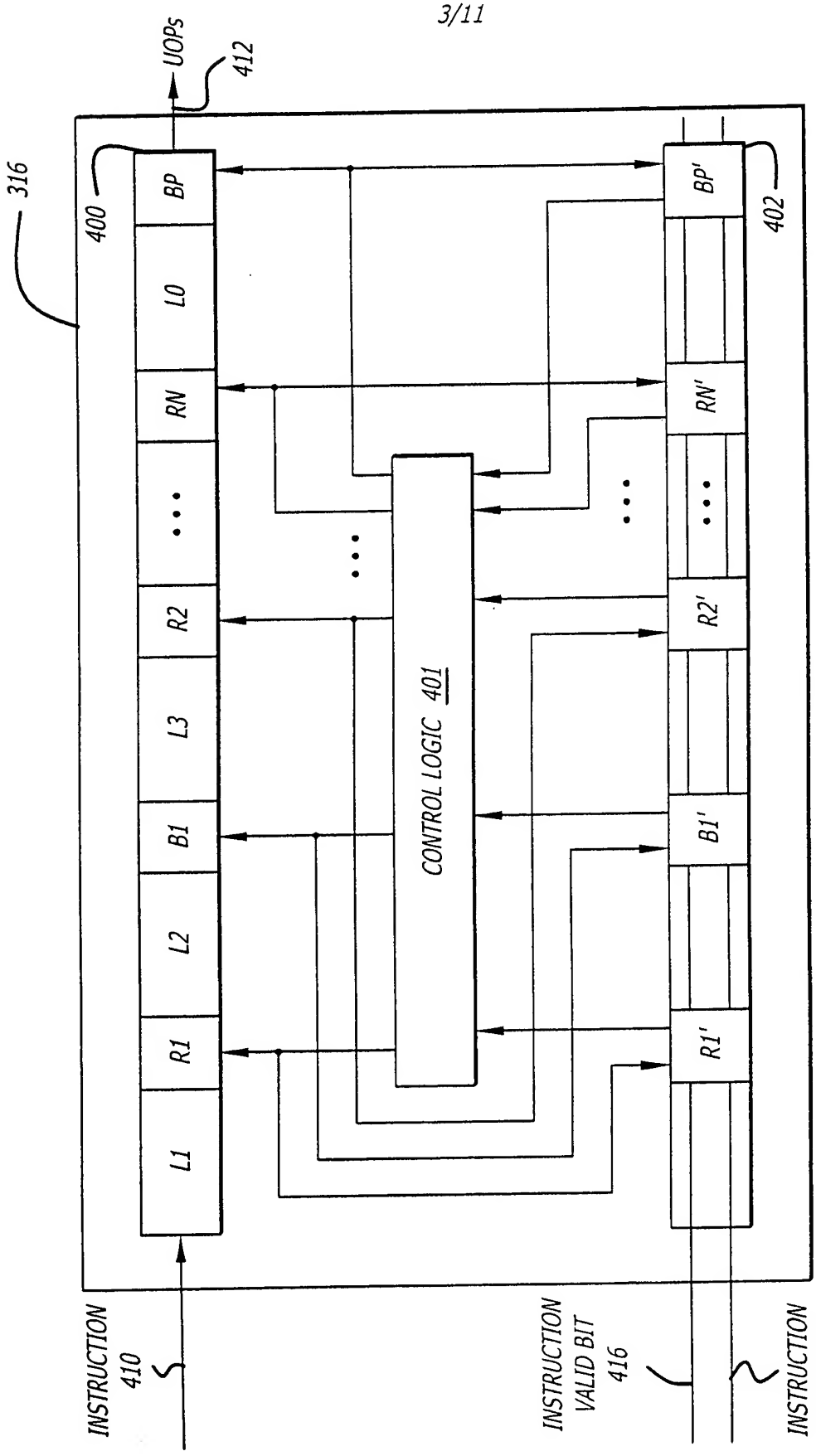
UOPs
412
316
400
402
BP
L0
RN
R2
L3
B1
L2
R1
L1
BP'
RN'
R2'
B1'
R1'
CONTROL LOGIC 401
INSTRUCTION 410
INSTRUCTION VALID BIT 416
INSTRUCTION THREAD ID 418

FIG. 4

4/11

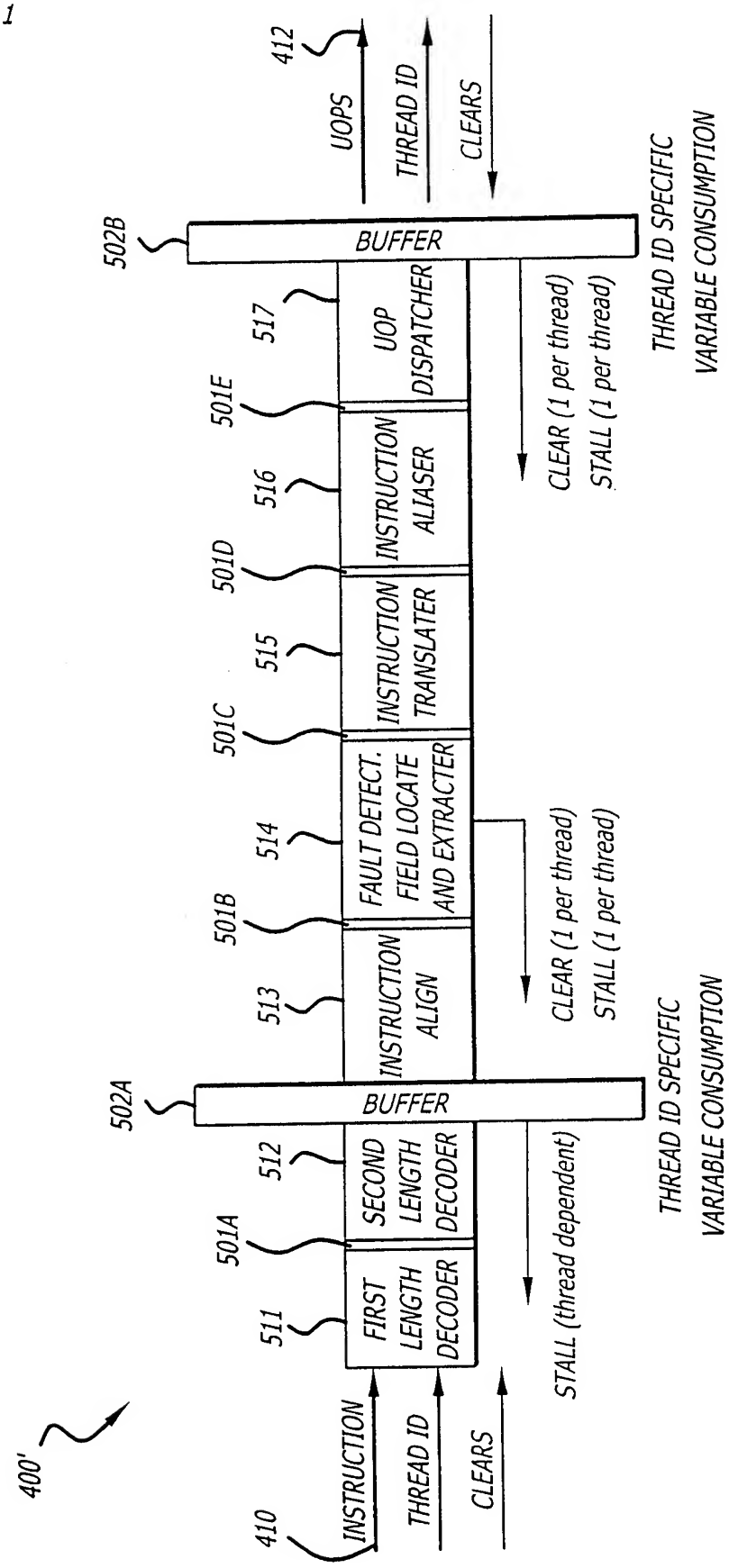
400'
410
INSTRUCTION
THREAD ID
CLEARS
511
FIRST LENGTH DECODER
501A
512
SECOND LENGTH DECODER
502A
BUFFER
513
INSTRUCTION ALIGN
501B
514
FAULT DETECT. FIELD LOCATE AND EXTRACTER
501C
515
INSTRUCTION TRANSLATER
501D
516
INSTRUCTION ALIASER
501E
517
UOP DISPATCHER
502B
BUFFER
412
UOPS
THREAD ID
CLEARS
STALL (thread dependent)
CLEAR (1 per thread)
STALL (1 per thread)
CLEAR (1 per thread)
STALL (1 per thread)
THREAD ID SPECIFIC VARIABLE CONSUMPTION
THREAD ID SPECIFIC VARIABLE CONSUMPTION

FIG. 5

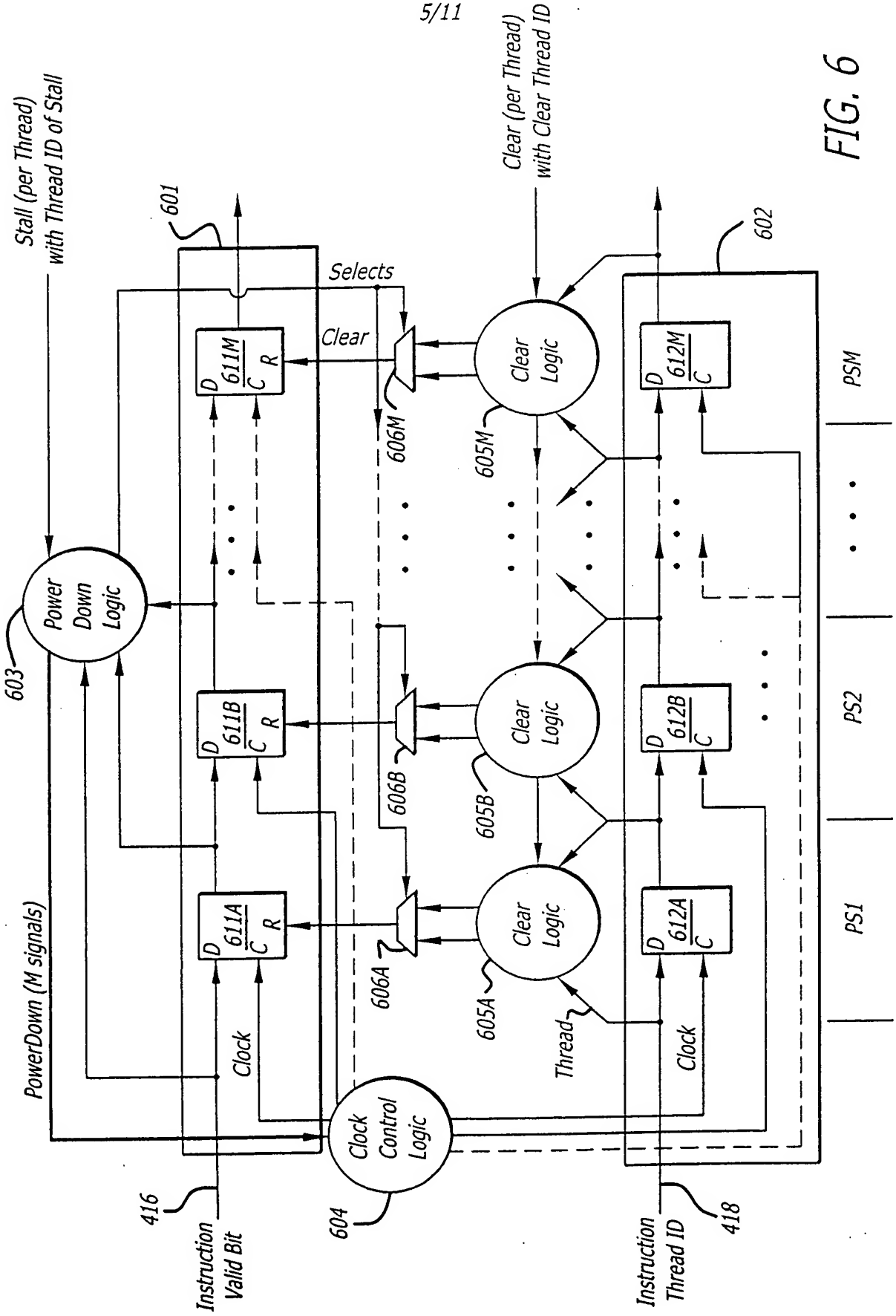
Stall (per Thread) with Thread ID of Stall
Clear (per Thread) with Clear Thread ID
601
602
Selects
Clear
611M
612M
606M
605M
Clear Logic
PSM
Power Down Logic
603
611B
612B
606B
605B
PS2
PS1
611A
612A
606A
605A
PowerDown (M signals)
Clock
Thread
Clock Control Logic
604
416
418
Instruction Valid Bit
Instruction Thread ID

FIG. 6

Stall for Next to Last PipeStage (NLP)

$Stall_{(NLP)}$ = *Valid Instruction in* $Pipe_{(NLP)}$ *AND* $(ThreadId_{(NLP)} = ThreadId\ of\ Stall)$

Stall for any other PipeStageX

$Stall_{(X)}$ = *Valid Instruction in* $Pipe_{(X)}$ *AND Valid Instruction in* $Pipe_{(X+1)}$ *AND* $Stall_{(NLP)}$

Powerdown for any PipeStage X

$Powerdown_{(X)}$ = *NOT Valid Instruction in* $Pipe_{(X-1)}$

Clock Enable for any PipeStage X

$Clock_{(X)}$ = *NOT* $Stall_{(X)}$ *AND NOT* $Powerdown_{(X)}$

Clock for any PipeStage X

$Clear_{(X)} = Clock_{(X)}$ *AND* $[(ClearThread_{(Id0)}$ *AND* $(ThreadId_{(X-1)} = Id0))$ *OR* $(ClearThread_{(Id1)}$ *AND* $(ThreadId_{(X-1)} = Id1))]$
OR
$NOT\ Clock_{(X)}$ *AND* $[(ClearThread_{(Id0)}$ *AND* $(ThreadId_{(X)} = Id0))$ *OR* $(ClearThread_{(Id1)}$ *AND* $(ThreadId_{(X)} = Id1))]$

$ClearThread_{(Id0)}$ = *There was a Clear on Thread Identification 0*

$ClearThread_{(Id1)}$ = *There was a Clear on Thread Identification 1*

Pipe(X) = Any pipestage in the decode
Pipe(X-1) = Pipestage before Pipe(X)
Pipe(X+1) = Pipestage after Pipe(X)

FIG. 7

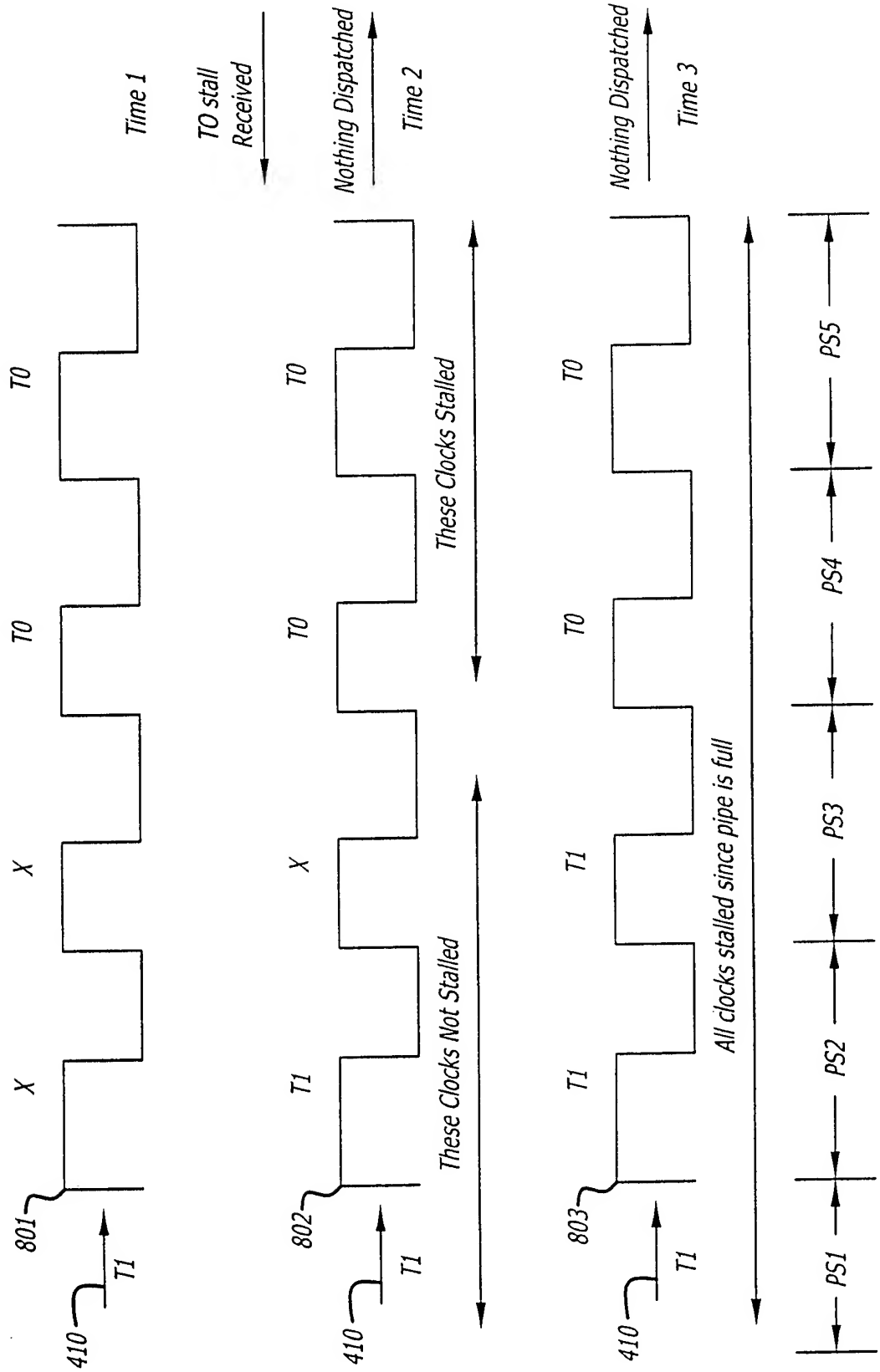
Time 1
TO stall Received
Nothing Dispatched
Time 2
Nothing Dispatched
Time 3
T0
T0
X
X
801
T1
410
T0
T0
X
T1
802
T1
410
These Clocks Stalled
These Clocks Not Stalled
T0
T0
T1
T1
803
T1
410
All clocks stalled since pipe is full
PS1
PS2
PS3
PS4
PS5

FIG. 8

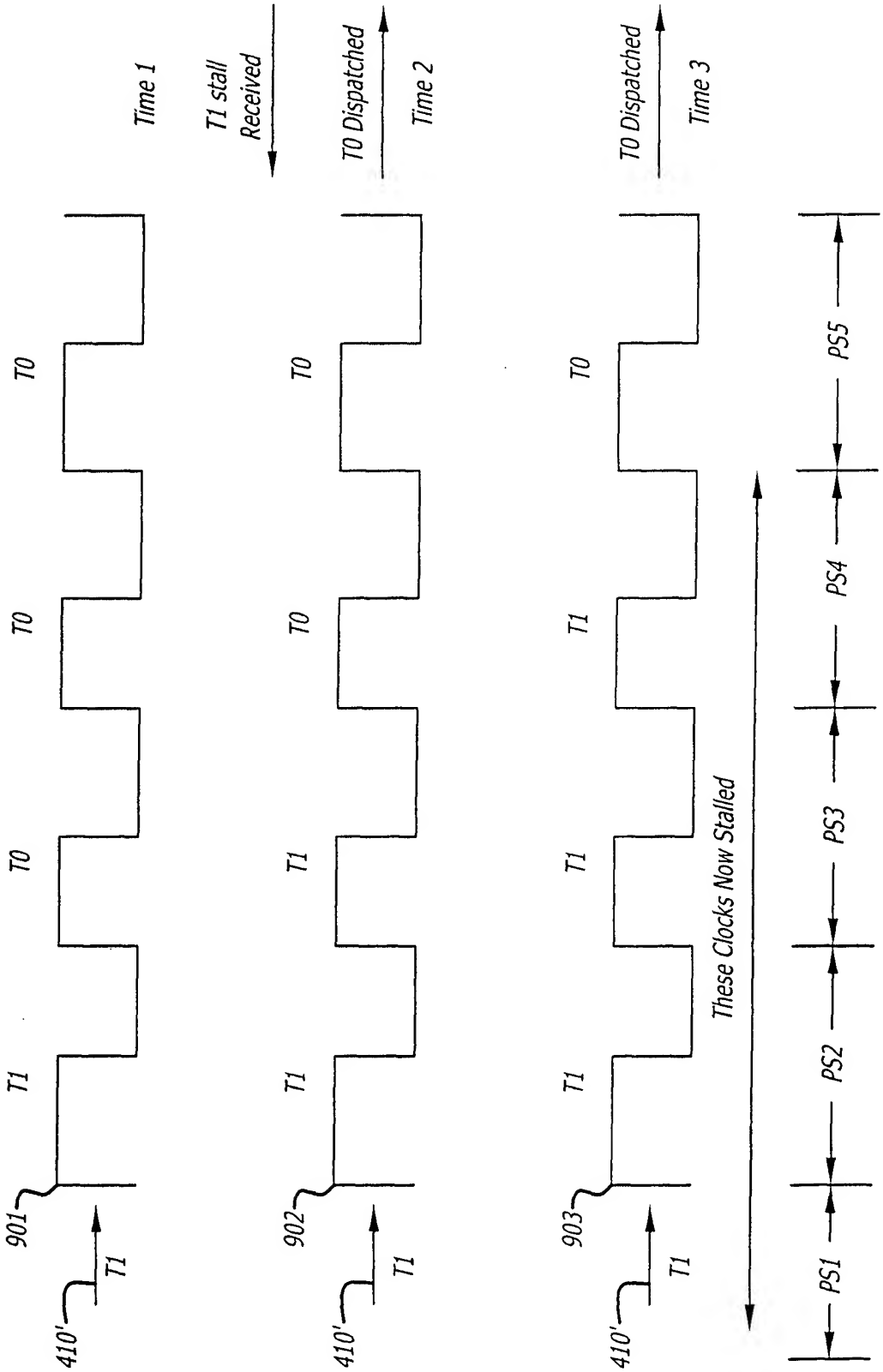
Time 1
T1 stall Received
T0 Dispatched
Time 2
T0 Dispatched
Time 3
T0
T0
T0
T1
901
410'
T1
T0
T1
902
410'
T1
T0
T1
T1
903
410'
T1
These Clocks Now Stalled
PS1
PS2
PS3
PS4
PS5

FIG. 9

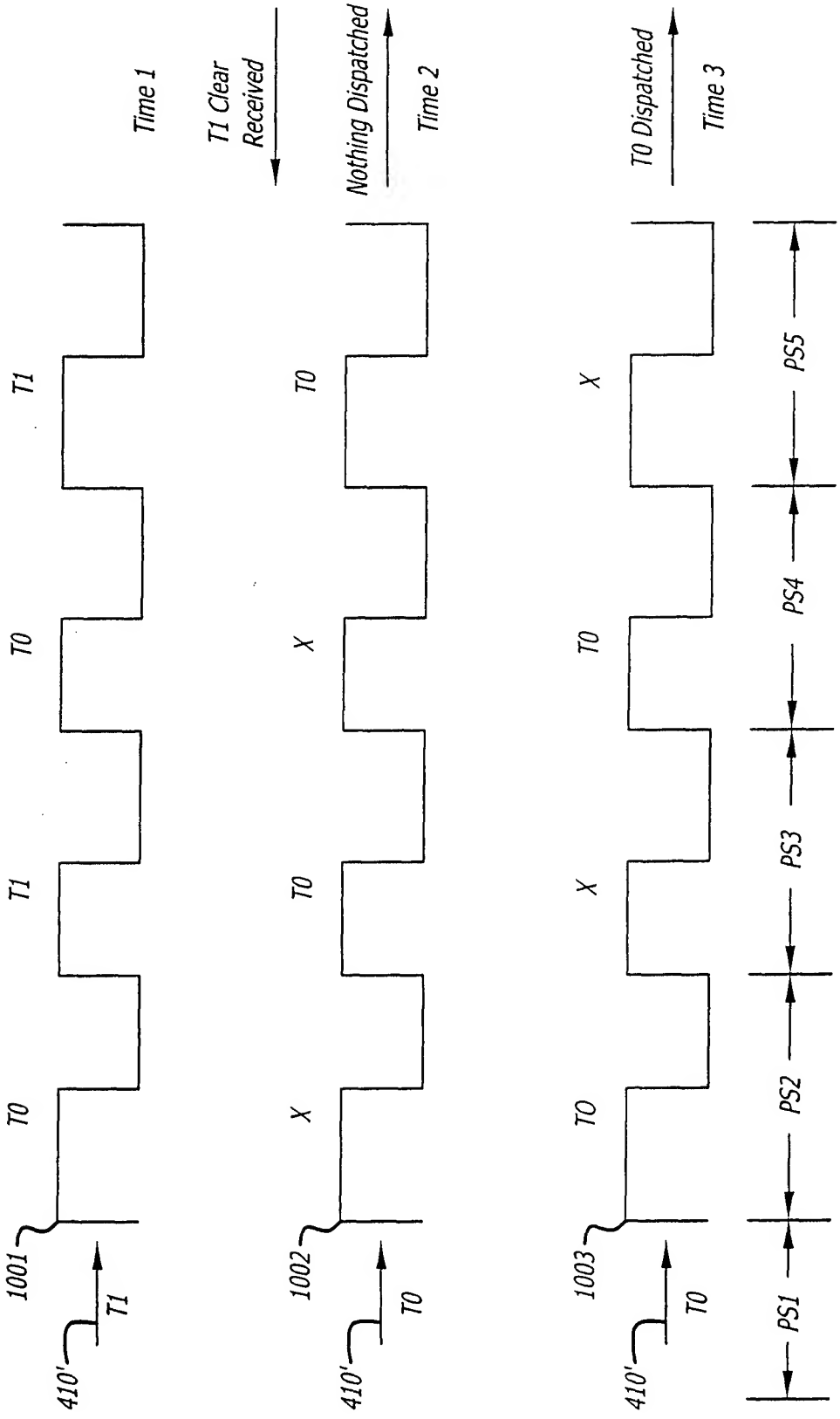
Time 1
T1 Clear Received
Nothing Dispatched
Time 2
T0 Dispatched
Time 3
T1
T0
T1
T1
T0
X
T0
X
T0
X
T0
1001
1002
1003
410'
T1
T0
T0
PS1
PS2
PS3
PS4
PS5

FIG. 10

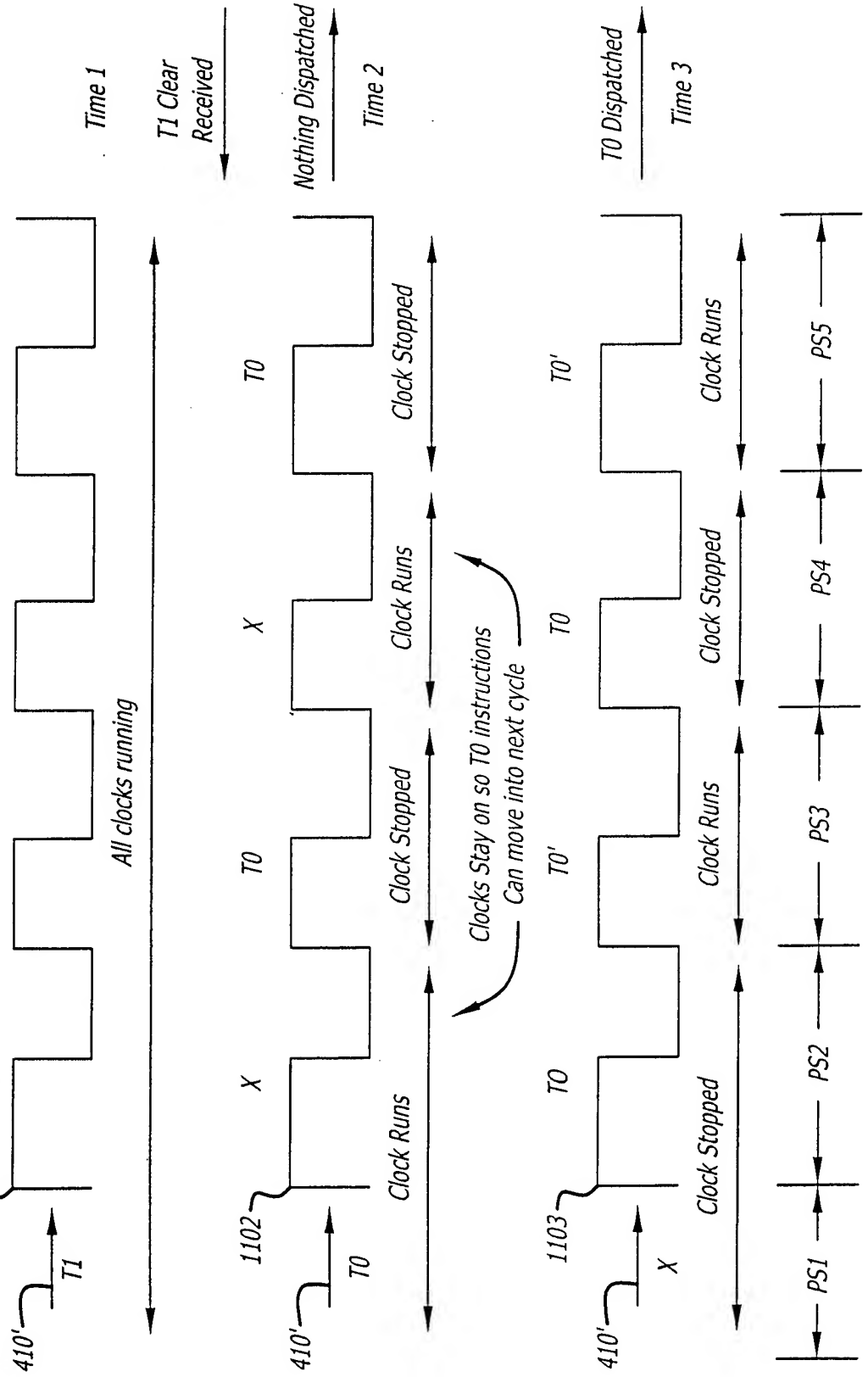
Time 1
T1 Clear Received
Nothing Dispatched
Time 2
T0 Dispatched
Time 3
All clocks running
Clocks Stay on so T0 instructions Can move into next cycle
Clock Runs
Clock Stopped
1101
1102
1103
410'
T1
T0
X
T0'
PS1
PS2
PS3
PS4
PS5

FIG. 11A

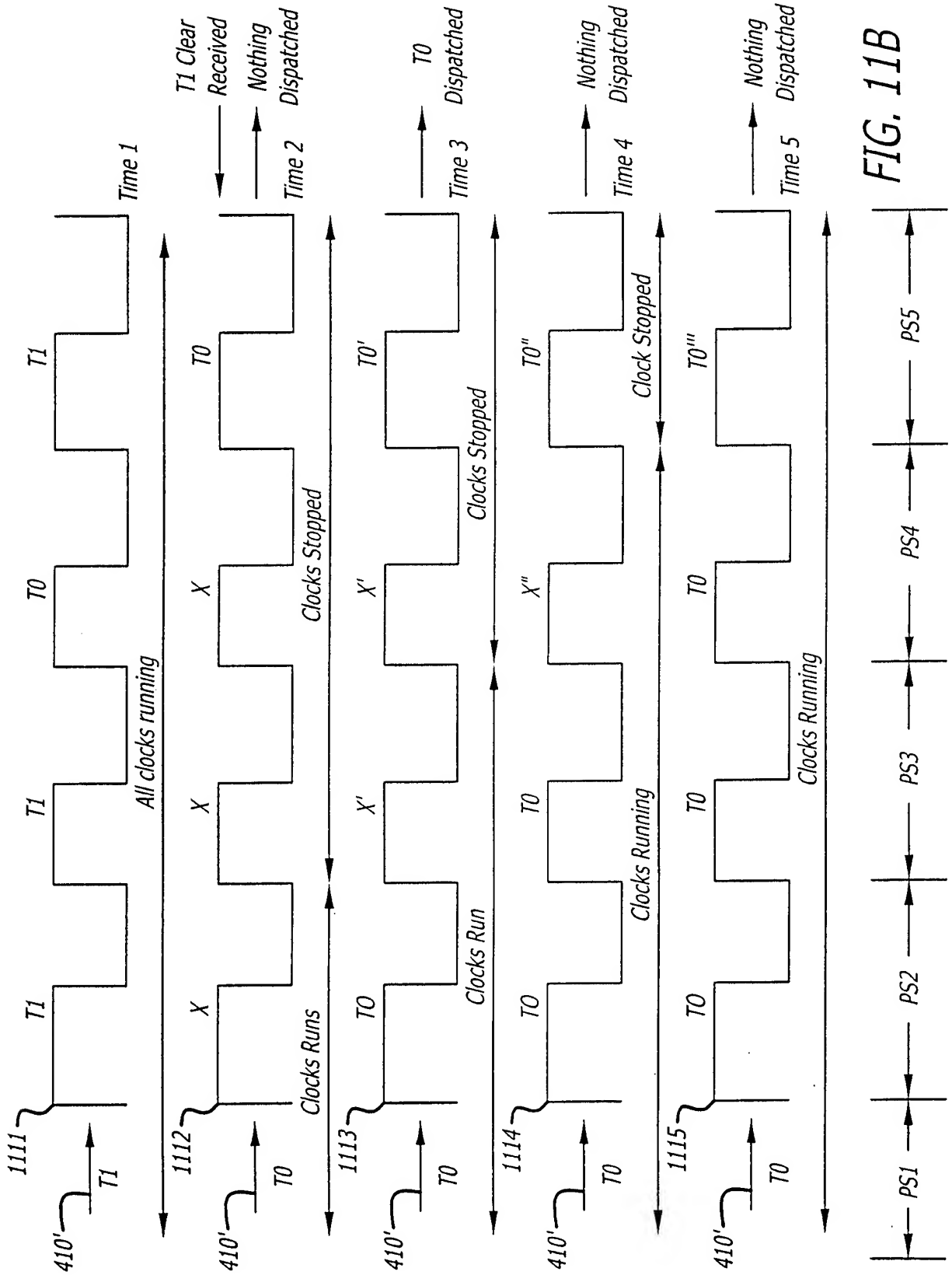
Time 1
Time 2
Time 3
Time 4
Time 5
T1 Clear Received
Nothing Dispatched
T0 Dispatched
Nothing Dispatched
Nothing Dispatched
All clocks running
Clocks Runs
Clocks Stopped
Clocks Run
Clocks Stopped
Clocks Running
Clock Stopped
Clocks Running
1111
1112
1113
1114
1115
410'
T1
T0
X
X'
X''
T0'
T0''
T0'''
PS1
PS2
PS3
PS4
PS5

FIG. 11B